U.S Accidents Project

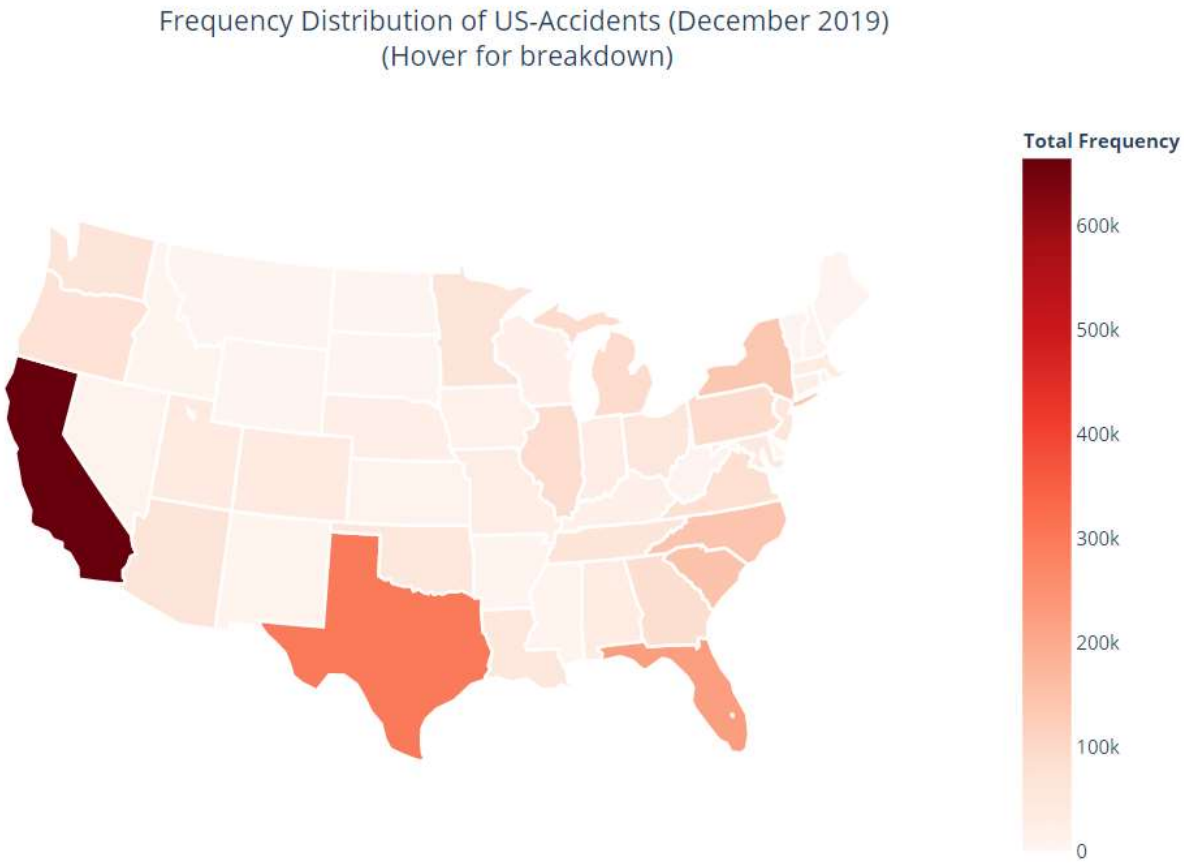
Thanh Duong, Henry Tran

1. Introduction

The data collected is a countrywide car accident dataset, which covers 49 states of the USA. The accident data are collected from February 2016 to June 2020. including two APIs which provide streaming traffic event data. These APIs broadcast traffic events captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Currently, there are about 3.5 million accident records in this dataset.

2. Understanding data

The dataset covers 49 states of the US. Following diagram shows the current data distribution over all the states.



The data is provided in terms of a CSV file. Following table describes the data attributes:

#	Attribute	Description	Nullable
1	ID	This is a unique identifier of the accident record.	No

2	Source	Indicates source of the accident report (i.e. the API which reported the accident.).	No
3	TMC	A traffic accident may have a Traffic Message Channel (TMC) code which provides more detailed description of the event.	Yes
4	Severity	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).	No
5	Start_Time	Shows start time of the accident in local time zone.	No
6	End_Time	Shows end time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow was dismissed.	No
7	Start_Lat	Shows latitude in GPS coordinate of the start point.	No
8	Start_Lng	Shows longitude in GPS coordinate of the start point.	No
9	End_Lat	Shows latitude in GPS coordinate of the end point.	Yes
10	End_Lng	Shows longitude in GPS coordinate of the end point.	Yes
11	Distance(mi)	The length of the road extent affected by the accident.	No
12	Description	Shows natural language description of the accident.	No
13	Number	Shows the street number in address field.	Yes
14	Street	Shows the street name in address field.	Yes
15	Side	Shows the relative side of the street (Right/Left) in address field.	Yes
16	City	Shows the city in address field.	Yes
17	County	Shows the county in address field.	Yes
18	State	Shows the state in address field.	Yes
19	Zipcode	Shows the zipcode in address field.	Yes
20	Country	Shows the country in address field.	Yes
21	Timezone	Shows timezone based on the location of the accident (eastern, central, etc.).	Yes
22	Airport_Code	Denotes an airport-based weather station which is the closest one to location of the accident.	Yes
23	Weather_Timestamp	Shows the time-stamp of weather observation record (in local time).	Yes
24	Temperature(F)	Shows the temperature (in Fahrenheit).	Yes
25	Wind_Chill(F)	Shows the wind chill (in Fahrenheit).	Yes
26	Humidity(%)	Shows the humidity (in percentage).	Yes
27	Pressure(in)	Shows the air pressure (in inches).	Yes
28	Visibility(mi)	Shows visibility (in miles).	Yes

29	Wind_Direction	Shows wind direction.	Yes
30	Wind_Speed(mph)	Shows wind speed (in miles per hour).	Yes
31	Precipitation(in)	Shows precipitation amount in inches, if there is any.	Yes
32	Weather_Condition	Shows the weather condition (rain, snow, thunderstorm, fog, etc.)	Yes
33	Amenity	A POI annotation which indicates presence of amenity in a nearby location.	No
34	Bump	A POI annotation which indicates presence of speed bump or hump in a nearby location.	No
35	Crossing	A POI annotation which indicates presence of crossing in a nearby location.	No
36	Give_Way	A POI annotation which indicates presence of give_way in a nearby location.	No
37	Junction	A POI annotation which indicates presence of junction in a nearby location.	No
38	No_Exit	A POI annotation which indicates presence of no_exit in a nearby location.	No
39	Railway	A POI annotation which indicates presence of railway in a nearby location.	No
40	Roundabout	A POI annotation which indicates presence of roundabout in a nearby location.	No
41	Station	A POI annotation which indicates presence of station in a nearby location.	No
42	Stop	A POI annotation which indicates presence of stop in a nearby location.	No
43	Traffic_Calming	A POI annotation which indicates presence of traffic_calming in a nearby location.	No
44	Traffic_Signal	A POI annotation which indicates presence of traffic_signal in a nearby location.	No
45	Turning_Loop	A POI annotation which indicates presence of turning_loop in a nearby location.	No
46	Sunrise_Sunset	Shows the period of day (i.e. day or night) based on sunrise/sunset.	Yes
47	Civil_Twilight	Shows the period of day (i.e. day or night) based on civil twilight.	Yes
48	Nautical_Twilight	Shows the period of day (i.e. day or night) based on nautical twilight.	Yes
49	Astronomical_Twilight	Shows the period of day (i.e. day or night) based on astronomical twilight.	Yes

3. Cleaning data and explore data

In this section, we use python to load this dataset, and try to clean data. This step will make the data more cleaning and ready for analysis.

- Use describe function in python, we have the first look of all numeric data:

```
train_df = pd.read_csv(r"D:\Data Download\US_Accidents_June20.csv")
train_df.ID.count()
train_df.describe()
```

	TMC	Severity	Start_Lat	Start_Lng	End_Lat	End_Lng
count	2.48E+06	3.51E+06	3.51E+06	3.51E+06	1.03E+06	1.03E+06
mean	2.08E+02	2.34E+00	3.65E+01	-9.58E+01	3.76E+01	-1.00E+02
std	2.08E+01	5.52E-01	4.88E+00	1.74E+01	4.86E+00	1.85E+01
min	2.00E+02	1.00E+00	2.46E+01	-1.25E+02	2.46E+01	-1.24E+02
25%	2.01E+02	2.00E+00	3.36E+01	-1.17E+02	3.40E+01	-1.18E+02
50%	2.01E+02	2.00E+00	3.59E+01	-9.10E+01	3.78E+01	-9.70E+01
75%	2.01E+02	3.00E+00	4.03E+01	-8.09E+01	4.11E+01	-8.21E+01
max	4.06E+02	4.00E+00	4.90E+01	-6.71E+01	4.91E+01	-6.71E+01

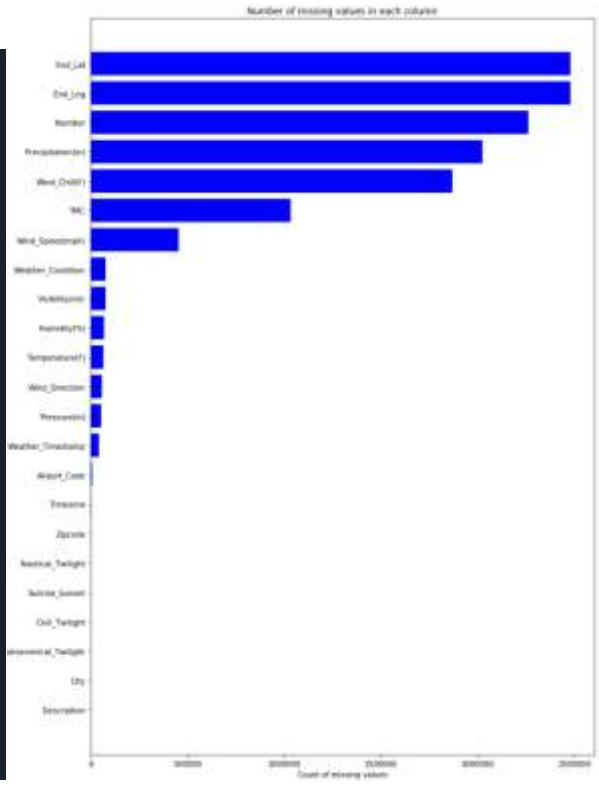
	Distance(mi)	Temperature(F)	Wind_Chill(F)	Humidity(%)	Pressure(in)	Visibility(mi)
count	3.51E+06	3.45E+06	1.65E+06	3.44E+06	3.46E+06	3.44E+06
mean	2.82E-01	6.19E+01	5.36E+01	6.51E+01	2.97E+01	9.12E+00
std	1.55E+00	1.86E+01	2.38E+01	2.28E+01	8.32E-01	2.89E+00
min	0.00E+00	#######	########	1.00E+00	0.00E+00	0.00E+00
25%	0.00E+00	5.00E+01	3.57E+01	4.80E+01	2.97E+01	1.00E+01
50%	0.00E+00	6.40E+01	5.70E+01	6.70E+01	3.00E+01	1.00E+01
75%	1.00E-02	7.59E+01	7.20E+01	8.40E+01	3.01E+01	1.00E+01
max	3.34E+02	1.71E+02	1.15E+02	1.00E+02	5.77E+01	1.40E+02

- Check null/missing data:

```
missing_df = train_df.isnull().sum(axis=0).reset_index()
missing_df.columns = ['column_name','missing_count']
missing_df = missing_df[missing_df['missing_count']>0]
missing_df = missing_df.sort_values(by='missing_count')

ind = np.arange(missing_df.shape[0])
width = 0.5
fig,ax = plt.subplots(figsize=(12,18))
rects = ax.barh(ind,missing_df.missing_count.values,color='blue')
ax.set_yticks(ind)
ax.set_yticklabels(missing_df.column_name.values, rotation='horizontal')
ax.set_xlabel("Count of missing values")
ax.set_title("Number of missing values in each column")
plt.show()
```

And we get a lot of column has null value:

column_name	missing_count
Description	1
City	112
Astronomical_Twilight	115
Civil_Twilight	115
Sunrise_Sunset	115
Nautical_Twilight	115
Zipcode	1069
Timezone	3880
Airport_Code	6758
Weather_Timestamp	43323
Pressure(in)	55882
Wind_Direction	58874
Temperature(F)	65732
Humidity(%)	69687
Visibility(mi)	75856
Weather_Condition	76138
Wind_Speed(mph)	454609
TMC	1034799
Wind_Chill(F)	1868249
Precipitation(in)	2025874
Number	2262864
End_Lng	2478818
End_Lat	2478818



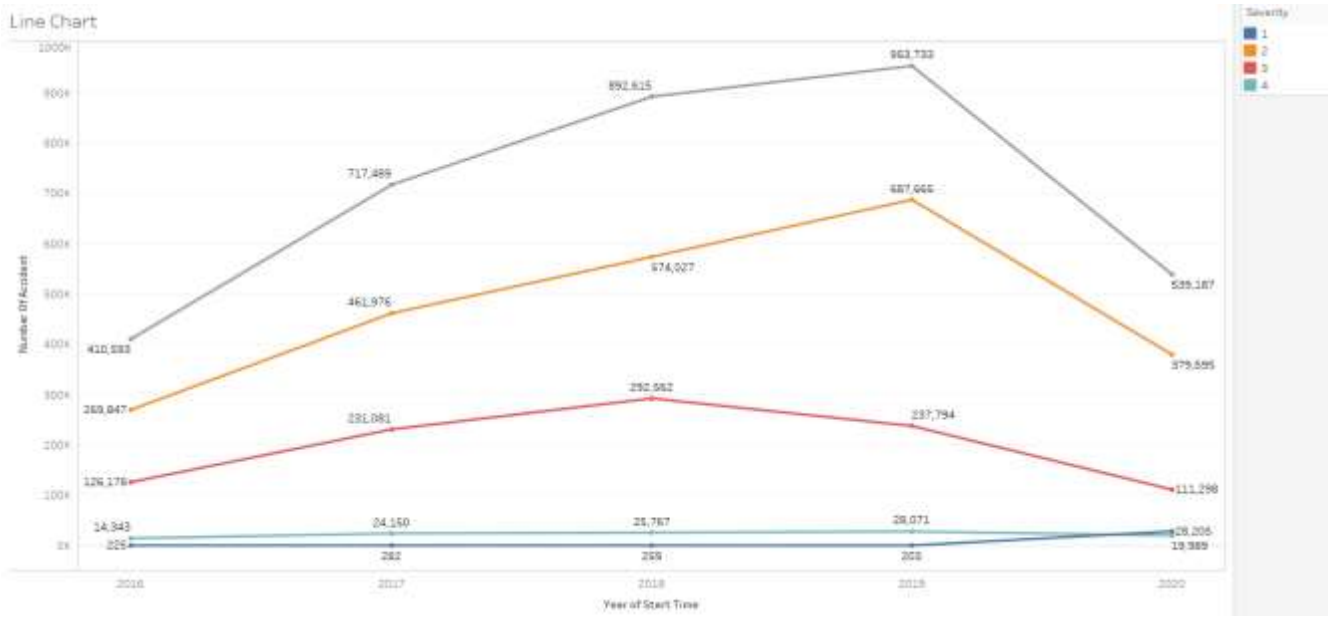
For column with 30% more null data on the total, we can't use to get the inside, the trend of data. The column TMC, Wind_Chill(F), Precipitation(in), Number, End_Lng, End_Lat will be removed.

4. Analysis Data

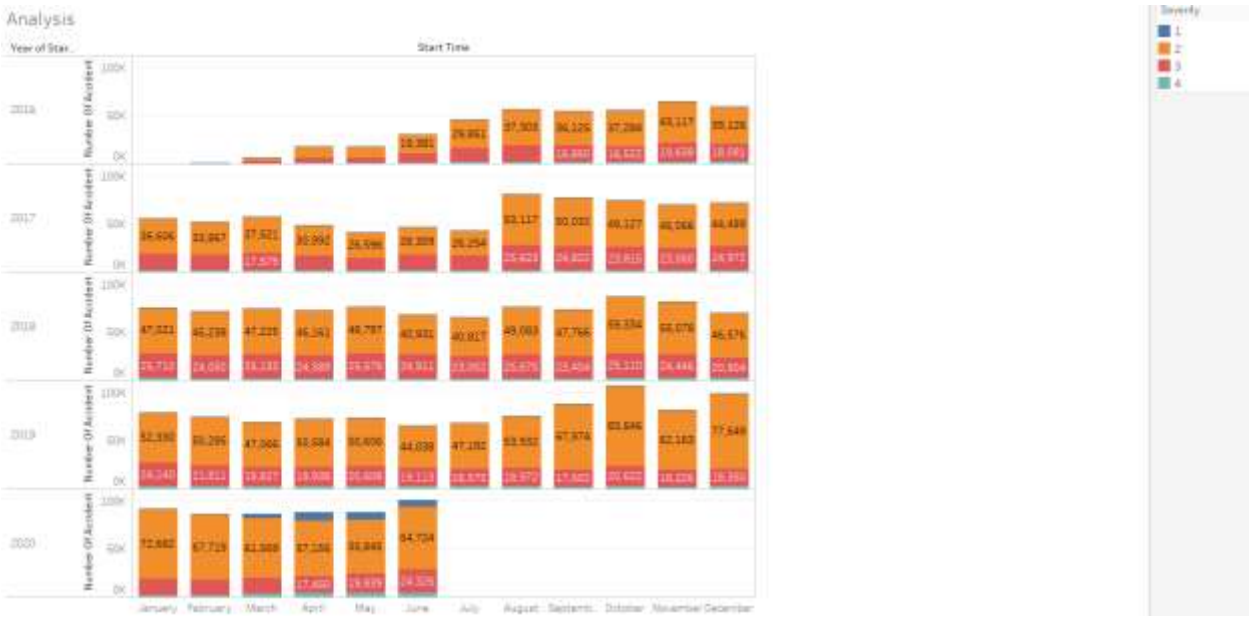
What is the trend of number of accidents from USA?

By breaking down this data by severity of the accident as the capture below, we can see:

- The grey line indicates the total of accidents which are increase year by year
- The orange indicates the accidents with severity 2 (Medium) which have the same trend.
- The remaining have the small proportion and not much change over the year.

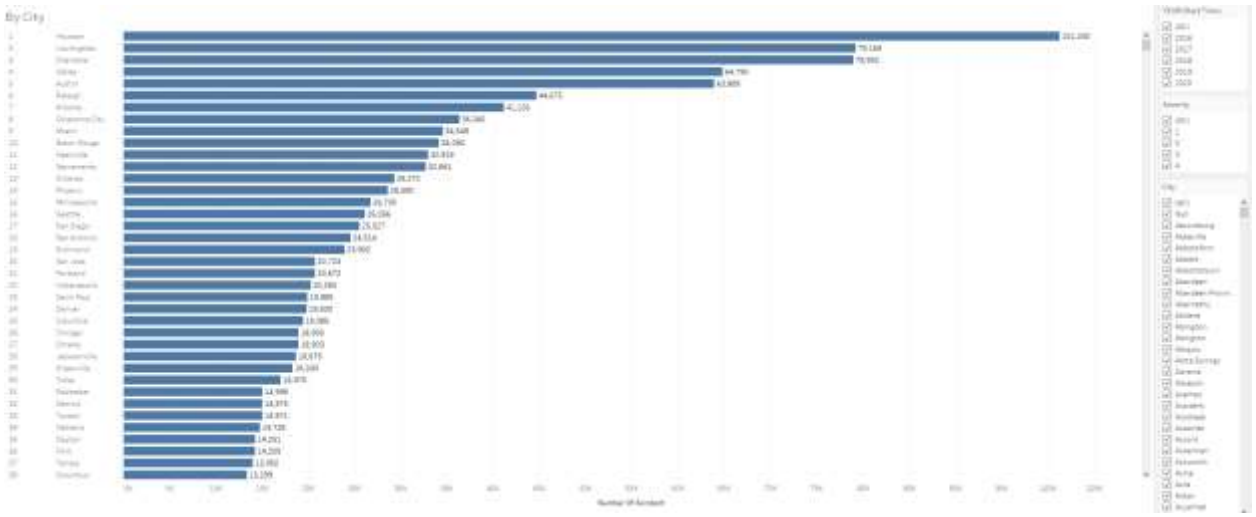


The accident with severity 2 and 3 have the most proportion. Althought the number accident has increased year by year, but the proportion of the serious accidents has tended to decrease. That a good signal.

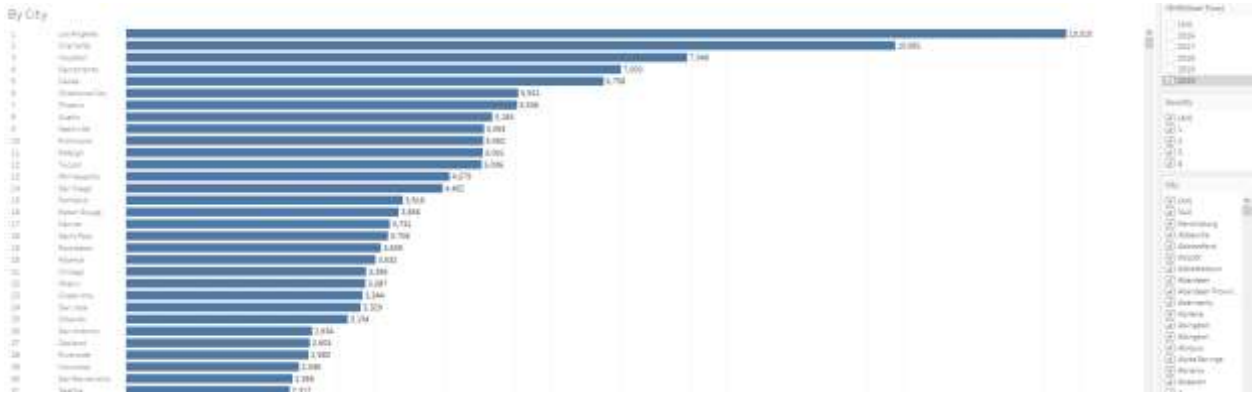


What is city having the highest number of accidents?

From 2016 to Jun 2020-06, Houston have the highest number with 101240 accidents.

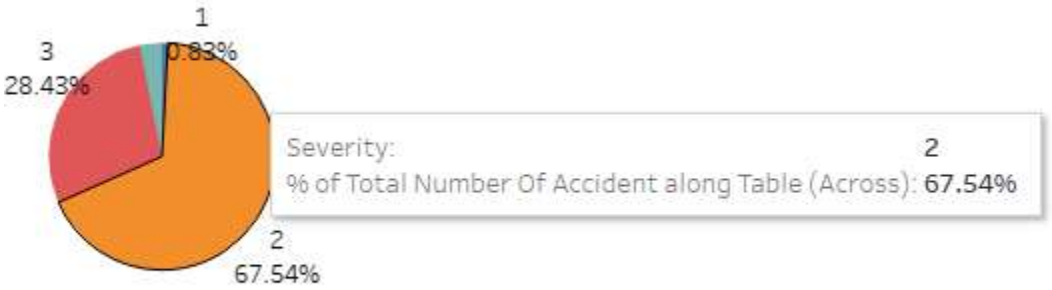


But only in 2020, Los Angeles is now having the highest accidents as the capture below:

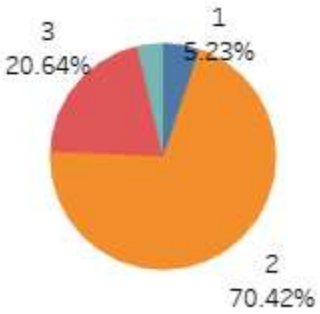


How is the distribution of accident by severity?

On the total data, 68% of number of accidents was recorded at level 2.

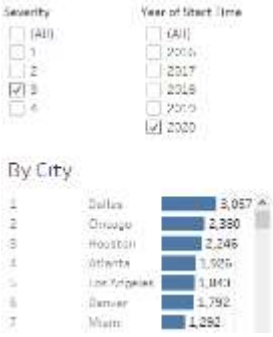


On 2020, The accidents with severity 2 increase but the severity 3 and 4 are decreased.



By compare the severity of accidents between cities:

At the most serious level (Severity 4), Dallas has the highest number of accidents, and followed by Washinton and Miami



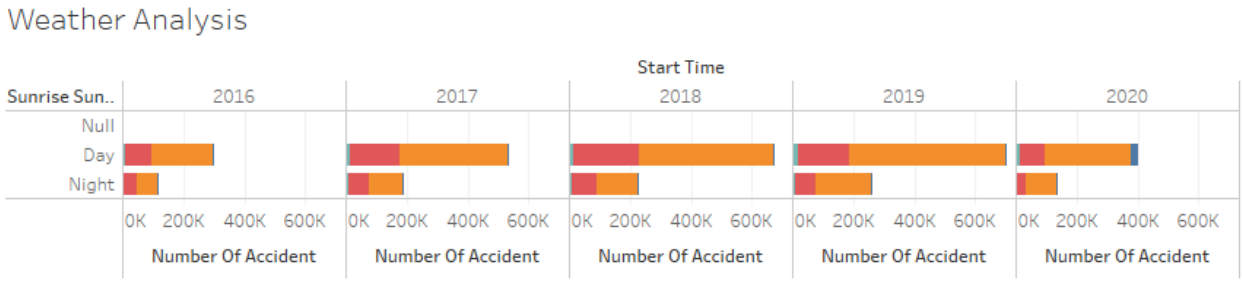
At the severity 3, Dallas is also the city with the highest number of accidents, followeb by Chicago, Houston, Atlanta.

At the level 2, Los Angeles, Chalotte, Sacramento are the cities with have highest accidents.

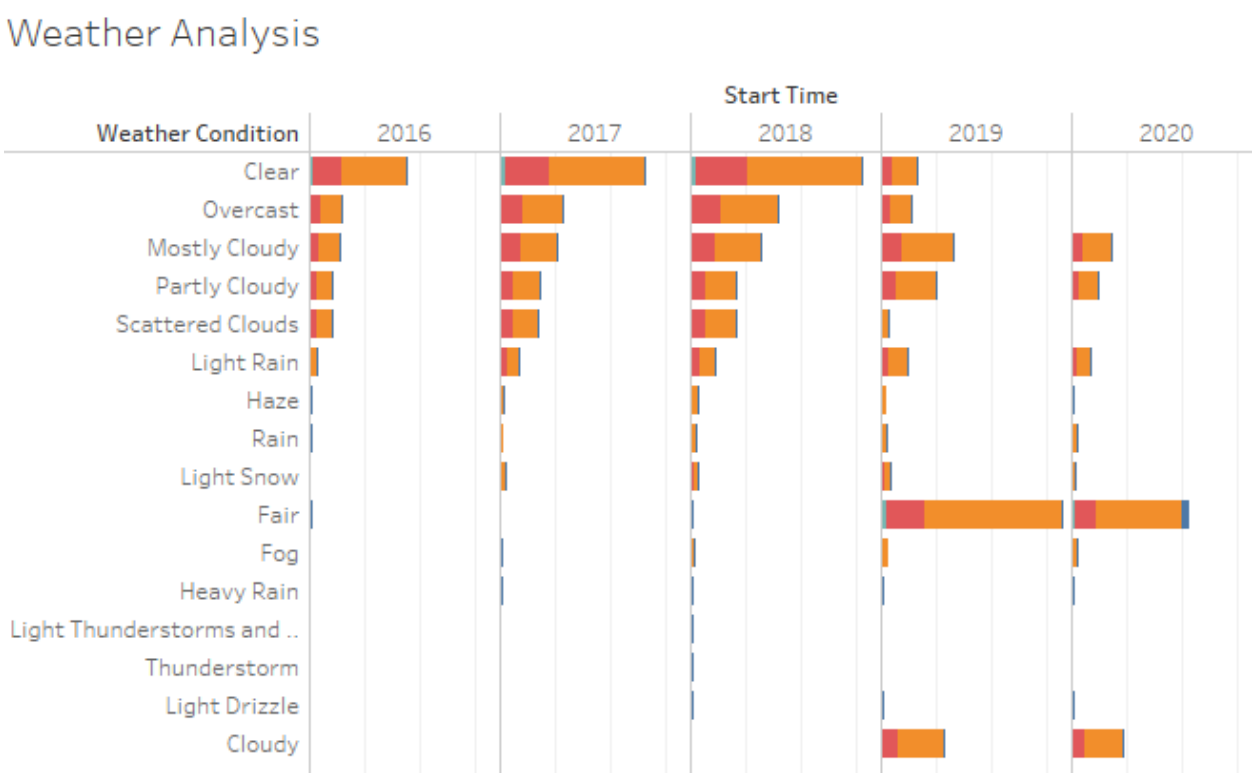
And the least serious accident, Tucson have the highest, followed by Phoenix and Richmond

How about the weather impact on the severity of accident?

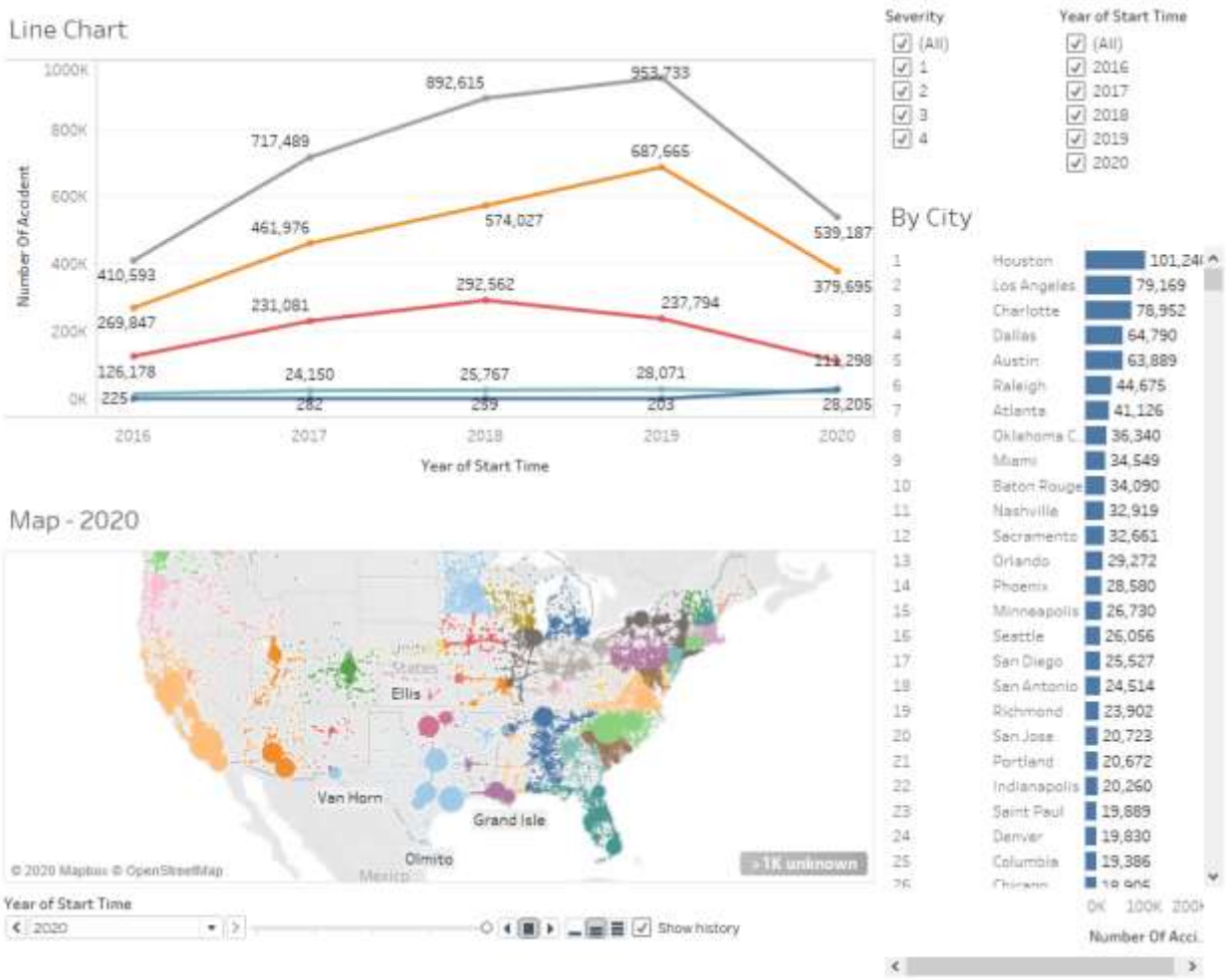
- Breakdown by Sunrise sunset, we can see the accident occur almost at day.



- Breakdown by Weather Condition, Clear and Fair condition appear almost in the accident, those are favorable weather condition for drivers, picnics, ….


Weather Analysis
Start Time
Weather Condition
2016
2017
2018
2019
2020
Clear
Overcast
Mostly Cloudy
Partly Cloudy
Scattered Clouds
Light Rain
Haze
Rain
Light Snow
Fair
Fog
Heavy Rain
Light Thunderstorms and ..
Thunderstorm
Light Drizzle
Cloudy

5. Dashboard



6. Apply Machine Learning

In this project, we apply Machine Learning model to predict the severity of accident base on the condition of weather, the actual weather at the time of accidents like temperature, wind direction, sunrise, sunset, …

2 models will be applied are Logistic Regression and Random Forest. The AUC will be used to determine the accuracy of the model and based on this AUC; we can decide whether we can use this model to detect the severity of the accidents.

The predictive results can be used to make it easier for the police to grasp the situation to allocate reasonable human resources to solve problems.

6.1. Prepare data

The variables are chosen for this Machine Learning Project:

Fields	Explanation
ID	Use the original data
Severity_Map	We keep 2 severity and map with the original data as below: Severity =1,2 -> Severity = 0 (Less Seriously)

	Severity =3,4 -> Severity = 1 (More Seriously)
Start_Time_Segment	We divide the Start_Time into 4 segment of time of the days: 0H-6H -> Map to 1 6H-12H -> Map to 2 12H-18H -> Map to 3 18H-24H -> Map to 4
End_Time_Segment	We divide the End_Time into 4 segment of time of the days: 0H-6H -> Map to 1 6H-12H -> Map to 2 12H-18H -> Map to 3 18H-24H -> Map to 4
Start_lat	Use the original data
Start_Lng	Use the original data
State	Use the original data
Temperature(F)	Use the original data
Wind_Chill(F)	Use the original data
Humidity(%)	Use the original data
Pressure(in)	Use the original data
Visibility(mi)	Use the original data
Wind_Direction	Use the original data
Wind_Speed(mph)	Use the original data
Precipitation(in)	Use the original data
Weather_Condition	Use the original data
Sunrise_Sunset	Use the original data

We use python to transform and prepare this data

- Load data

```
#Load Data
train_df = pd.read_csv(r"D:\Data Download\US_Accidents_June20.csv")

df = train_df[['ID','Severity', 'Start_Time',
               'End_Time','Start_Lat','Start_Lng',
               'State','Temperature(F)','Wind_Chill(F)',
               'Humidity(%)','Pressure(in)','Visibility(mi)',
               'Wind_Direction','Wind_Speed(mph)','Precipitation(in)',
               'Weather_Condition','Sunrise_Sunset'
              ]]
```

- Create “Severity_Map” column

```
# Map Severity to 0,1
df.loc[df['Severity'] <= 2, 'Severity_Map'] = '0'
df.loc[df['Severity'] >  2, 'Severity_Map'] = '1'
```

- Create “Start_Time_Segment” column

```
# Cast data to datetime
df['Start_Time']= pd.to_datetime(df['Start_Time'])
df['End_Time']= pd.to_datetime(df['End_Time'])

#Segment Start Time
df['Start_Time_Hour'] = df['Start_Time'].dt.hour.map("{:02}".format).astype(int)

df.loc[(df['Start_Time_Hour'] >= 0) & (df['Start_Time_Hour'] < 6), 'Start_Time_Segment']    = '0H-6H'
df.loc[(df['Start_Time_Hour'] >= 6) & (df['Start_Time_Hour'] < 12), 'Start_Time_Segment']   = '6H-12H'
df.loc[(df['Start_Time_Hour'] >= 12) & (df['Start_Time_Hour'] < 18), 'Start_Time_Segment']  = '12H-18H'
df.loc[(df['Start_Time_Hour'] >= 18) & (df['Start_Time_Hour'] < 24), 'Start_Time_Segment']  = '18H-24H'
```

- Create “End_Time_Segment” column

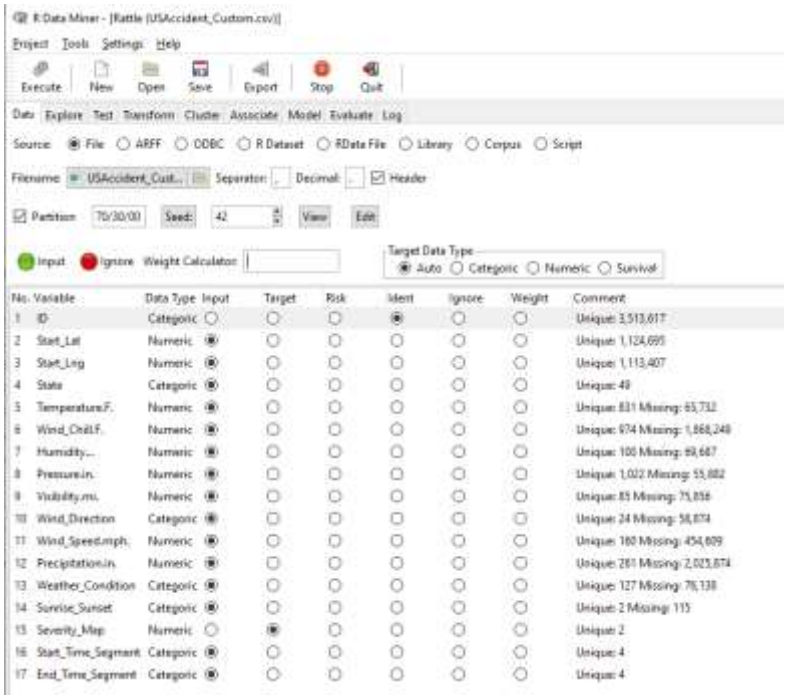
```
#Segment End Time
df['End_Time_Hour'] = df['End_Time'].dt.hour.map("{:02}".format).astype(int)

df.loc[(df['End_Time_Hour'] >= 0) & (df['End_Time_Hour'] < 6), 'End_Time_Segment']    = '0H-6H'
df.loc[(df['End_Time_Hour'] >= 6) & (df['End_Time_Hour'] < 12), 'End_Time_Segment']   = '6H-12H'
df.loc[(df['End_Time_Hour'] >= 12) & (df['End_Time_Hour'] < 18), 'End_Time_Segment']  = '12H-18H'
df.loc[(df['End_Time_Hour'] >= 18) & (df['End_Time_Hour'] < 24), 'End_Time_Segment']  = '18H-24H'
```

- To minimize the data, we choose data in 2016 to build the Machine Learning Model

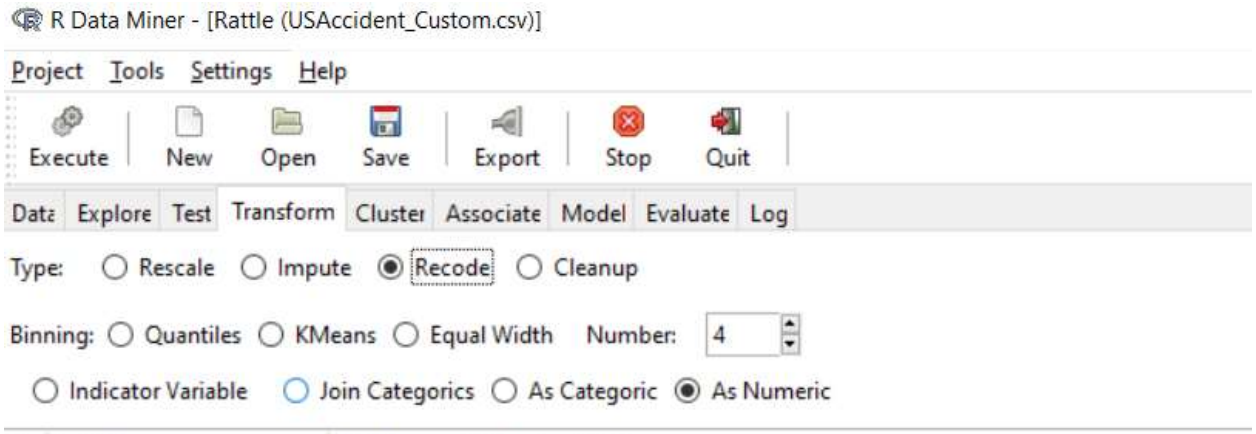
```
df_2016= df[df['Start_Time'].dt.year == 2016]
```

6.2. Apply Random Forest

Step 1: We use Rattle Tools to build this model, firstly, we import the prepared data in the step above, and set Severity_Map as the target of project:

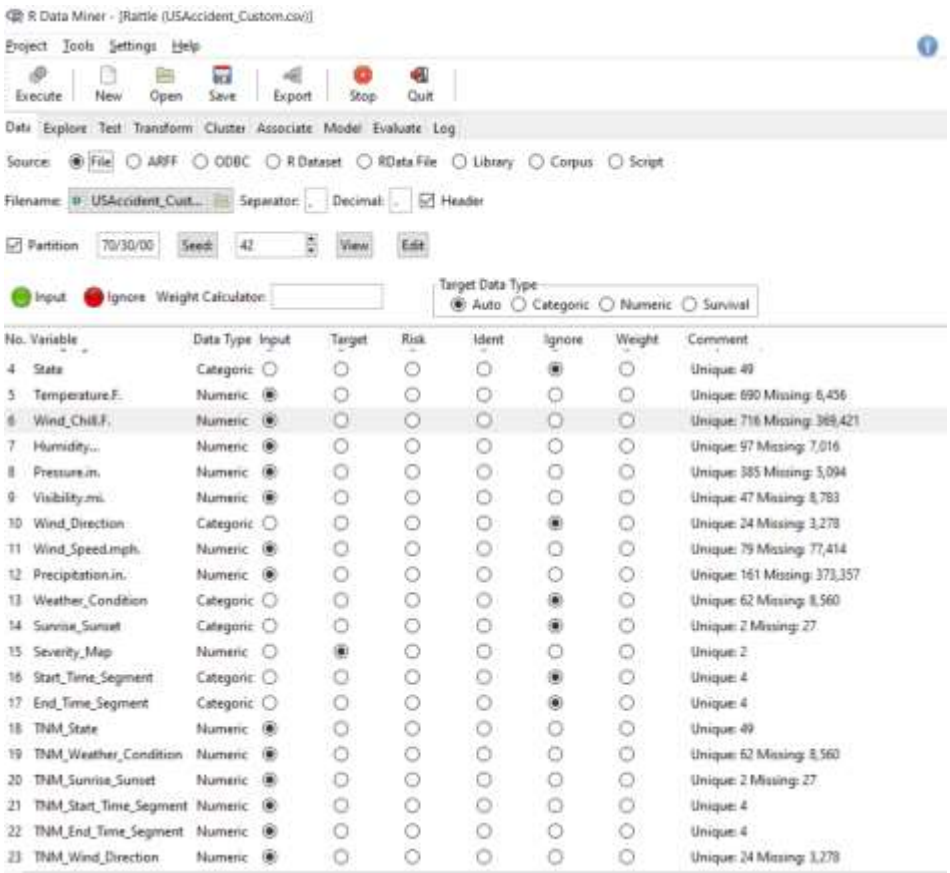


Step 2: Convert all category data to number by using recode function in Rattle tools

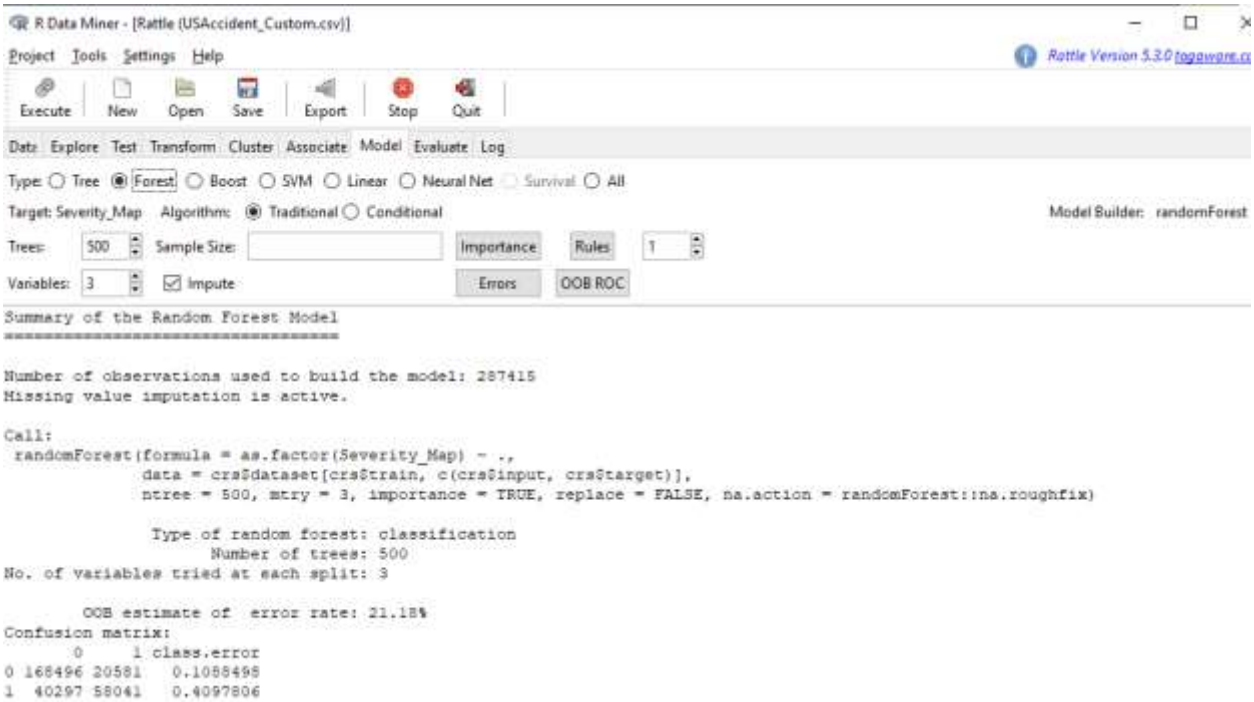


No.	Variable	Data Type and Number Missing
1	ID	Categorical [410593 levels].
2	Start_Lat	Numeric [25.16 to 48.99; unique=188658; mean=36.29; median=36.20].
3	Start_Lng	Numeric [-124.42 to -67.85; unique=187399; mean=-97.71; median=-95.43].
4	State	Categorical [49 levels; ignored].
5	Temperature.F.	Numeric [-20.20 to 161.60; unique=690; mean=67.04; median=69.10; miss=6456].
6	Wind_Chill.F.	Numeric [-41.50 to 111.00; unique=716; mean=31.55; median=31.80; miss=369421].
7	Humidity...	Numeric [4.00 to 100.00; unique=97; mean=63.71; median=65.00; miss=7016].
8	Pressure.in.	Numeric [0.12 to 33.04; unique=385; mean=30.01; median=30.01; miss=5094].
9	Visibility.mi.	Numeric [0.00 to 111.00; unique=47; mean=9.39; median=10.00; miss=8783].
10	Wind_Direction	Categorical [24 levels; miss=3278; ignored].
11	Wind_Speed.mph.	Numeric [0.00 to 822.80; unique=79; mean=8.52; median=8.10; miss=77414].
12	Precipitation.in.	Numeric [0.00 to 10.14; unique=161; mean=0.06; median=0.01; miss=373357].
13	Weather_Condition	Categorical [62 levels; miss=8560; ignored].
14	Sunrise_Sunset	Categorical [2 levels; miss=27; ignored].
15	Severity_Map	Numeric [0 to 1; unique=2; mean=0; median=0].
16	Start_Time_Segment	Categorical [4 levels; ignored].
17	End_Time_Segment	Categorical [4 levels; ignored].
18	TNM_State	Numeric [1.00 to 49.00; unique=49; mean=19.05; median=13.00].
19	TNM_Weather_Condition	Numeric [1.00 to 62.00; unique=62; mean=24.68; median=32.00; miss=8560].
20	TNM_Sunrise_Sunset	Numeric [1.00 to 2.00; unique=2; mean=1.27; median=1.00; miss=27].
21	TNM_Start_Time_Segment	Numeric [1.00 to 4.00; unique=4; mean=2.88; median=3.00].
22	TNM_End_Time_Segment	Numeric [1.00 to 4.00; unique=4; mean=2.74; median=3.00].
23	TNM_Wind_Direction	Numeric [1.00 to 24.00; unique=24; mean=12.63; median=14.00; miss=3278].

Step 3: Construct model with new recode variable:

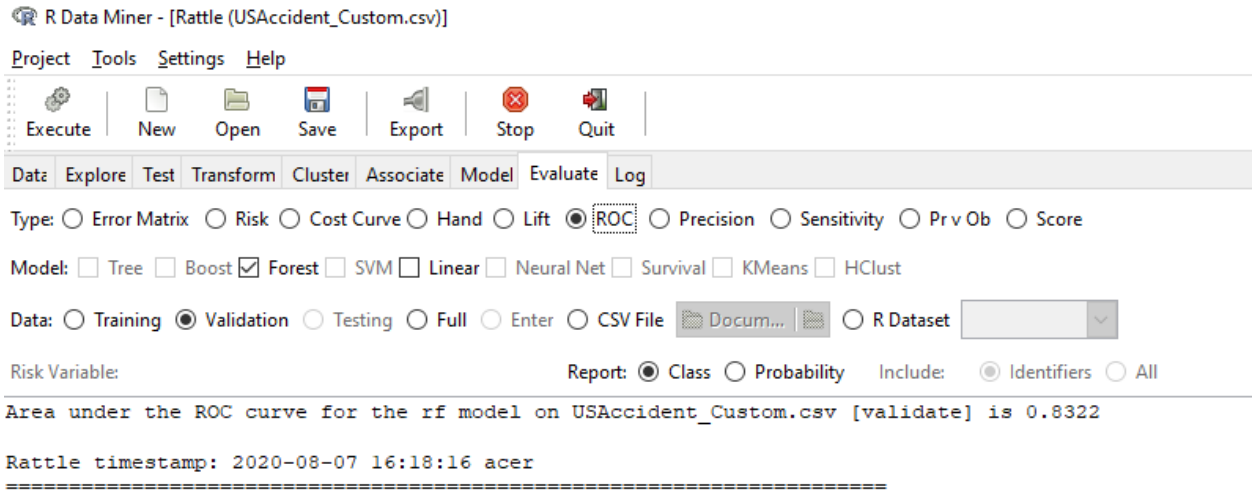


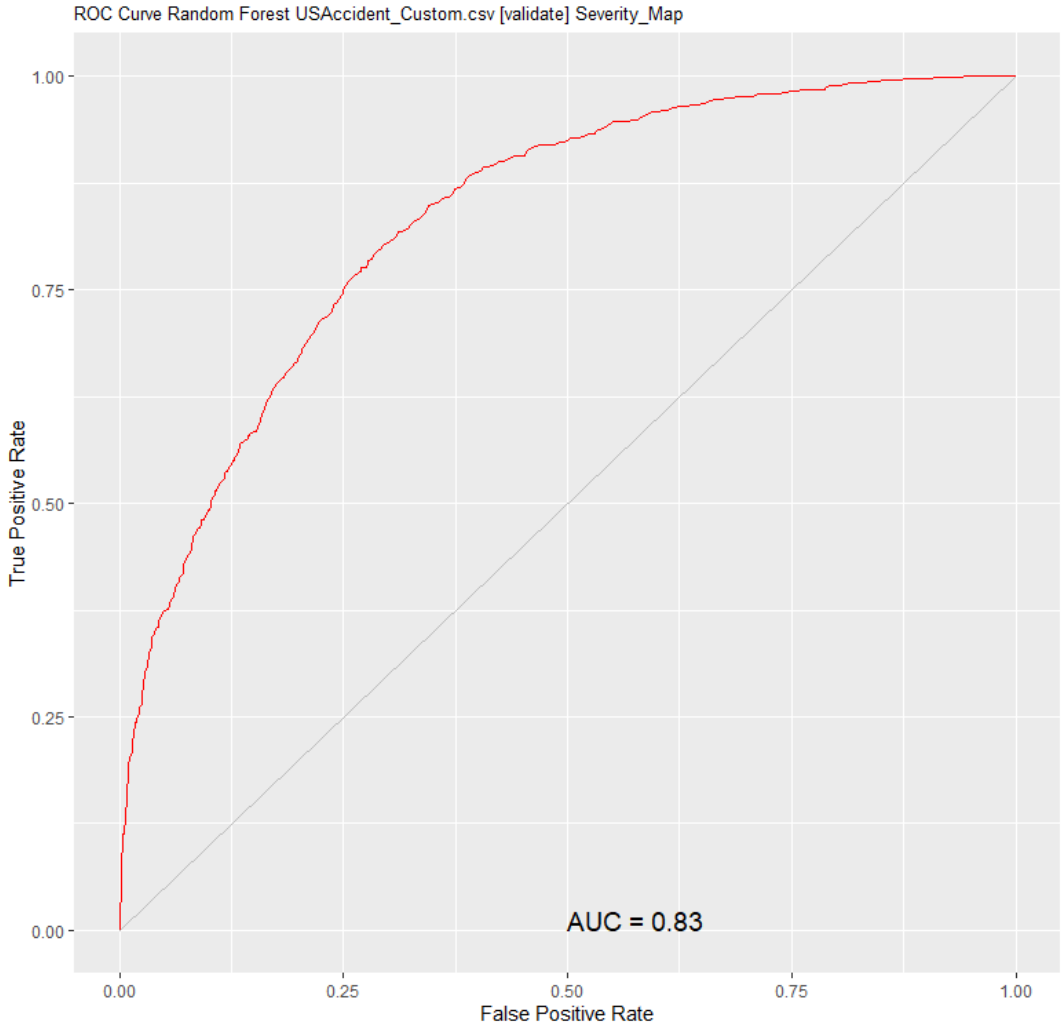
Step 4: Run Random Forest model in Rattle Tools



6.3. Model evaluation

We ROC Curves in the validation tab to evaluate the model





6.4. Conclusion

In this part, we use Random Forest to construct model to predict the severity of Accident. And we get the good AUC with 0.83. So, we can apply this model to fast predict the severity and the police can save time to assess and allocate reasonable human resources to solve problems.

7. Reference

https://www.kaggle.com/sobhanmoosavi/us-accidents

Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of Statistical Learning - Data Mining, Inference, and Prediction. Berlin: Springer-Verlag.
This is an excellent text that explains some of the key ideas in machine learning within a statistical framework.

Jordan, M. (2003). Probabilistic Graphical Models. Professor Jordan has kindly shared a pre-publication draft.
This text has an excellent coverage of generative and discriminative probabilistic models for classification.

Kearns, M. and Vazirani, U. (1994). Computational Learning Theory. Cambridge, MA: MIT Press.
This, although a bit dated, is an excellent introduction to learning theory.

Mitchell, T. (1997). Machine Learning. New York: Mc Graw-Hill.
This is, although a bit dated, an excellent introduction to Machine Learning.